Control programado con Arduino

Práctica 10: O Sistema Automatizado con Retorno de Datos

Chegaches á última parada da nosa viaxe de aprendizaxe! Nesta práctica final integraremos practicamente todo o que aprendiches. Deseñaremos un sistema de seguridade intelixente. Para estruturar o código de maneira impecable, crearemos unha función de diagnóstico que avaliará o estado do pulsador e devolverá un valor lóxico (`return`) ao bucle principal, decidindo se activa o relé de potencia e os LEDs de aviso.

Obxectivos de Aprendizaxe

[1pt]

Ao rematar esta sesión serás capaz de:

- Deseñar funcións personalizadas que procesen información e devolvan valores lóxicos mediante a instrución `return`.
- Comprender os tipos de retorno de datos en C++ (en especial, o tipo booleano `bool`).
- Integrar múltiples compoñentes de entrada e saída (pulsador, LEDs de sinalización e relé de potencia) nun único circuíto complexo.
- Desenvolver la lóxica dun sistema industrial de seguridade de parada de emerxencia.

Compoñentes Necesarios

[1pt]

Este é a montaxe máis completa do curso. Coloca os seguintes compoñentes no teu simulador TinkerCad:

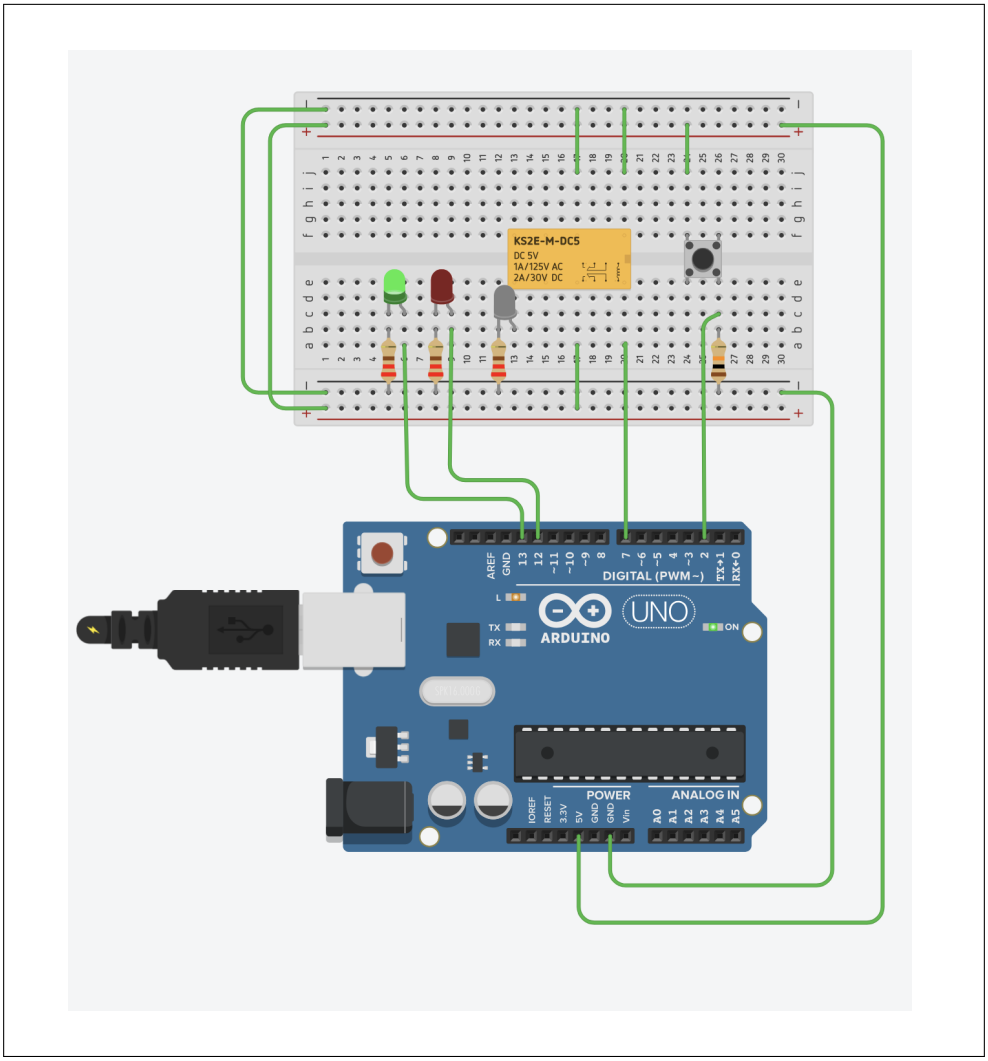
- 1 Placa Arduino Uno.
- 1 Placa de prototipado pequena (Protoboard).
- 2 Diodos LED (1 Verde de ”Sistema Seguro” e 1 Vermello de ”Perigo”).
- 2 Resistencias de $220\,\Omega$ (para protexer os LEDs).

- 1 Pulsador de catro patillas (Botón de emerxencia).
- 1 Resistencia de $10\,\text{k}\Omega$ (para a configuración Pull-Down do botón).
- 1 Relé SPDT de 5 Voltios (para simular o corte de alimentación de maquinaria de potencia).
- Cables de conexión virtuais.

Esquema de montaxe na Protoboard

[1pt]

Organiza o cableado con paciencia. Lembra que a bobina do relé se activa mediante un pin de Arduino, e os seus contactos gobernan a potencia de forma totalmente independente.



Código Base: Funcións con tipo de retorno 'bool'

[1pt]

Escribe o seguinte programa en TinkerCad. Observa que o tipo de datos da nosa función xa no é `void`, senón `bool`, o que significa que nos devolverá un valor de verdadeiro (`true`) ou falso (`false`) utilizando a orde `return`.

```
// Practica 10: Sistema de seguridade industrial con retorno de estado
int pinBoton = 2;
int pinRele = 7;
int ledVerde = 13;
int ledRojo = 12;

void setup() {
  pinMode(pinBoton, INPUT);
  pinMode(pinRele, OUTPUT);
  pinMode(ledVerde, OUTPUT);
  pinMode(ledRojo, OUTPUT);
}

void loop() {
  // 1. CHAMADA Á FUNCIÓN: Avaliamos o boton e gardamos o resultado
  bool alarmaActiva = evaluarSeguridad(pinBoton);

  // 2. ACCIÓNS EN BASE AO RETORNO DA FUNCIÓN
  if (alarmaActiva == true) {
    // Se a alarma esta activa, disparamos o rele e acendemos o LED vermello
    digitalWrite(pinRele, HIGH);
    digitalWrite(ledRojo, HIGH);
    digitalWrite(ledVerde, LOW);
  } else {
    // Se todo esta seguro, o rele descansa e o LED verde brilla
    digitalWrite(pinRele, LOW);
    digitalWrite(ledRojo, LOW);
    digitalWrite(ledVerde, HIGH);
  }
}

// 3. DEFINICIÓN DA FUNCIÓN CON RETORNO
// Indicamos 'bool' ao inicio porque a función debe devolver un booleano
bool evaluarSeguridad(int pinSensor) {
  int lectura = digitalRead(pinSensor);

  // Se se detecta unha pulsación (perigo detectado)
  if (lectura == HIGH) {
    return true;  // Devolvemos o valor 'verdadero' ao loop()
  } else {
    return false; // Devolvemos o valor 'falso' ao loop()
  }
}
```

Listing 1: Código utilizando unha función con retorno de datos para avaliar o sistema.

¿Cómo funciona el código y la instrución `return`? [1pt]

Ata agora, todas as nosas funcións terminaban coa palabra `void` porque só realizaban accións físicas. Pero en programación é vital que as funcións realicen cálculos ou lecturas e nos "devolvan" unha resposta.

- **Tipo de retorno (`bool`)**: Na cabeceira, substituímos `void` por `bool`. Isto obriga á función a devolver un dato booleano (verdadero ou falso) antes de pecharse.
- **La instrución `return`**: É a porta de saída da función. Envía o valor seleccionado de volta á liña exacta onde se chamou. En canto o programa le `return`, a función péchase de inmediato.
- **A recepción do dato**: Na liña `bool` , o Arduino executa a función e reemplaza todo ese bloque polo resultado devolto (`true` ou `false`), gardándoo directamente na variable local .

O reto da práctica 10: ¡El Desafío Final!

Consolida os teus coñecementos completando este gran reto de programación estruturada de seguridade:

1. **Alarma Acústica Temporalizada**: Engade un patrón de parpadeo de emerxencia. Se a función devolve `true`, fai que o LED Vermello parpadee rapidamente de forma continua, mentres que o LED Verde se manteña apagado.
2. **Función de Retorno Numérico**: Deseña unha nova función de diagnóstico chamada `comprobarErrores()` que non devolva un valor booleano, senón un número enteiro (`int`). Se o sistema está apagado, debe devolver `0`. Se o sistema se activa, debe contar os segundos que permanece activo e devolver ese valor numérico ao `loop()` para amosar o tempo de retardo na resposta.
3. *Pregunta para reflexionar*: Parabéns por completar as 10 prácticas! Escribe unha breve valoración persoal sobre como evolucionou a túa forma de programar desde la Práctica 1 (onde todo era secuencial) ata esta Práctica 10 (con funcións con paso de parámetros e retornos). Como te axuda esta estrutura a pensar como un/unha enxeñeiro/a de software?